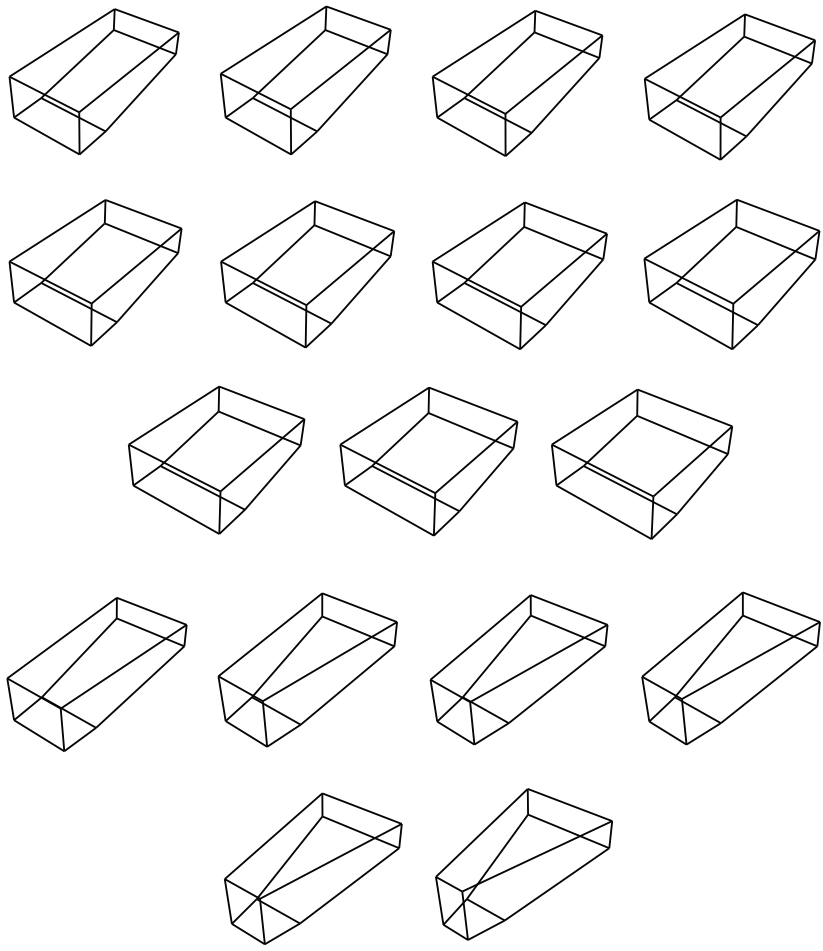
Computational Search in Architectural Design



Computational Search in Architectural Design

By Tomás Méndez Echenagucia

Computational Search in Architectural Design

By Tomás Méndez Echenagucia

.

Tomás Méndez Echenagucia

Computational Search in Architectural Design

Tutors: Mario Sassone · Pierre-Alain Croset · Arianna Astolfi

A thesis submitted for the degree of "Doctor of Philosophy"
XXVI cycle
2011 · 2012 · 2013
Ph.D. in "Architecture and Building Design"
Politecnico di Torino

.

Ph.D. in “Architecture and Building Design”
XXVI cycle
2011 · 2012 · 2013
Politecnico di Torino
Viale Mattioli 39
I-10125, Torino, Italy
Ph.D. candidate: Tomás Méndez Echenagucia
Registration number: 179121
Tutors: Mario Sassone, Pierre-Alain Croset and Arianna Astolfi

.

A Perucho.

Contents

Acknoledgements

Back in 2006 I had a first conversation with Maarten Jansen, Mario Sassone and Arianna Astolfi about what i wanted to do for my Master's thesis. In the following months they proceeded to turn a naively ambitious idea into serious research, they taught me to understand design challenges as serious research opportunities, to elevate a project to a learning experience. This PhD research is an inevitable consequence of that meeting and their generosity.

I first want to thank my tutors Pierre-Alain Croset, Mario Sassone and Arianna Astolfi, their constant work and guidance in their respective fields, their generosity and perseverance made them ideal mentors and companions in this research.

True multi-disciplinary work is impossible without constant collaboration between specialists, this research would not have been possible without the constant work of Louena Shtrepi, Arthur van der Harten, Alfonso Capozzoli and Ylenia Cascone. I will always be grateful for their contribution.

I received a great deal of help from my Master thesis students Marco Palma, Maddalena Sarotto, Sabrina Canale, Silvia Pastorino, Chiara Bertolutti and Denise Barbaroux.

My PhD colleagues from various departments were a great source of support and discussion, i would like to single out Dario Parigi, Andrea Rosada, Silvia Cammarano, Lorenza Bianco, Iasef Rian, Andrea Dutto, Paolo Tecchio, Shaghayegh Rajabzadeh, Giuseppina Puglisi, Antonio Spinelli, Ilaria Ariolfo, Matteo Malandrino and Enrico Boffa.

Professional work is most often not only the inspiration for this kind of architectural research, but quite often it is the source of great knowledge. Working in interesting projects gave me the curiosity and thirst for research that took me through this process. Priceless advice, discussions and practical knowledge was imparted to me by working with Maarten Jansen, Vanja Frlan, Mawari Nuñez, Daniel Otero, Alejandro Méndez, Kristian Ceballos,

Frans Swarte, Alina Delgadillo, Michel Cova, Marco Amosso, Domenico Ghirotto, Daniel Beckman, Motoo Komoda and Bob Mahoney.

Conversations with may people were also invaluable for me during these years, sometimes they would involve the research and practical questions, other times just a simple outlook on design, science or life. E-mails, talks or phone calls with the following people helped me a lot: Alesia Griginis, Alberto Pugnale, Elisa Cattaneo, Daniel Bosia, Steve Baer, Andreas Kloeckner, Isabella Rombi, Ricardo Avella, Ariadna Weisshar, Daniel Alvares, Gustavo Méndez, Juan Pablo Méndez, Argenis Lugo, Maria Eugenia Sosa, Domingo Acosta, Alfredo Cilento, Luis Rosales, Beatriz Hernandez, Winfried Lachenmayr, Eckard Kahle, Thomas Scelo, Tappio Lokki, Sakkari Tervo, Yann Yurkoviz, Martin Vercammen and Kirk Martini.

I also have a deep appreciation of the financial support given to me by the Politecnico di Torino. In these days if austerity i never took for granted the significance of this contribution.

The constant support of my parents though the years is the reason why I am here.

Introduction

The computer entered into the vast majority of architectural design studios by the mid 90's, but for many years it was used only as a replacement to the drafting table. Popular Computer Aided Design (CAD) applications, along with the plotter, were mostly a computational version of the same tools architects had been using for centuries. The aid they provided the designer was mostly in the drawing area, not too much in the design or construction fields. Improvements in this first phase of computation in the profession relate to speed and reliability, they do not represent significant functional additions to the architect's toolbox.

Coming from the mechanical industry, computational drafting tools have their origins in the post-war research projects of american universities, most notably the MIT. As is the case with many other appropriated technologies, their diffusion had to wait many years.

After the mid 90s, architects were pushed as many other professions, in the mainstream of the informatics revolution: software became more and more sophisticated and powerful, and the complexity of digital tools was very high.

Computer graphics advancements, as well as 3D modeling techniques, produced a series of computer applications destined to create 3D renderings. These applications were mostly intended for use in the cinema and entertainment fields, but they found their way into the architectural office soon enough. Architects were eager to communicate their ideas to their clients in a more direct and intuitive way. The images generated can surely achieve photo-realistic results in a way that was not possible with previous mediums, but again here, now new design or construction functionality was introduced. Applications that were destined for the entertainment industry however, were endowed with other characteristics that interested architects. Representation was a part of their use, but the modeling techniques used in this industry were more than mere geometrical objects, the models con-

tained hierarchical and invariant relationships. These features were helping the architects create and modify their 3D models in better ways.

Information began to find its way into the models as well. Architects began to use the digital model to generate and store information. Perhaps the most heard about use of the computer in the architectural office nowadays is the use of Building Information Modeling or BIM. BIM can be summed up as the combination of data bases with 3D CAD models, to form geometrical objects that contain all of the information generated during the design, as well as the necessary to construct a building. BIM applications are designed to contain graphical information of the design object in many scales, to parametrically modify its geometry, and to exchange information with other involved professionals in a very efficient way.

Software packages that contain building performance simulation also began to appear. Simulation software range from Finite Element calculations of structures to Computational Fluid Dynamic simulations of natural ventilation of indoor environments. These group of applications is used mostly by specialist consultants to the designers and not by the designers themselves, but it is also a significant addition to the architecture and construction field. The fast and precise modeling of physical phenomena concerning the technical performance of the building being designed is a good thing to have.

Software in the architecture field was a standardized product. Commercial software houses introduced a big amount of functionality in their products with the intention of capturing the attention of the market, but their tools were the same for all architects, all projects and all processes.

In the first decade of the XXI century the architectural community began to take consciousness of the deep impact that the digital revolution had, and architects started to investigate its implications. The "Architecture non-standard" exhibition held in 2004 at the centre Georges Pompidou in Paris, is the manifest of such renewed attention to the relationship between architects and their instruments and techniques. A group of architects started to gain interest in the creation of their own instruments and to become software developers. They began to respond more to their own interests and not let themselves be pushed into any particular tool. The creative efforts of architects found its way into the creation of digital tools with the intention of generating design solutions that were not possible before.

This new generation of designers customized commercial software for their own creative processes. Applications are no longer the same for all architects and all projects, custom processes required custom tools. Customization did not come about for representational problems, custom applications were made for the creative process.

The reliable storage, exchange and manipulation of precise building information is surely useful during the design process. But do these abilities help designers generate better design ideas? Simulation softwares provide performance data as an output to a given building geometry, materials and configuration. But does this information on its own help architects find a good way forward? An argument can be made to say that CAD, BIM and simulation softwares have so far been used mostly for the later stages of design. Little use has ben given to these tools in the conceptual design stage and this is because they are not intended for such use.

The early or conceptual stage of the design process is not so much about information storage, exchange or manipulation, and much more about information gathering. Early design is not helped by the ability to represent in great detail building components and specifications, it is much better assisted by exploration and evaluation.

Search Algorithms represent an opportunity for designers and specialists alike to generate architectural solutions that maximize performance values such as structural or energy efficiency. Building shapes and special features can be explored and high-performing features can be signaled out. By studying high-performing shapes, designers can learn about the relationships between shapes and many performance related disciplines. *Parametric Modeling* in combination with building simulation software and search algorithms allow designers to gather specific information on not just a particular solution, but on entire sets or families of geometric possibilities.

The objective of this PhD research is to investigate how best to use computational search processes in the early phase of design. Search algorithms are implemented in combination with parametric models of different kinds of geometry, with the purpose of studying various building performances.

The early phase of design and the nature of the design problem are studied to get an idea of the kind of search process that would best accompany architects during this phase. Multi-disciplinarity and contrasting objectives are singled out as fundamental required characteristics of such a process. This leads to the proposal of multi-disciplinary studies into architectural shapes, both in the realm of complex curved geometry and in more traditional orthogonal forms.

The search process itself is studied in its capacity to generate solutions in a multi-objective setting. The process of selecting and formulating search problems, the parametrization of geometric families for study as well as the selection criteria for outstanding solutions are all topics of discussion. A particular type of algorithm called Genetic Algorithms is implemented and studied in length. Search processes are proposed for 3 architectural design

fields: structural, acoustic and energy design.

This PhD thesis is divided in **three** parts. The first part is entitled "The concept of Search in Architectural Design" and it is presents the conceptual framework for this research. The second part is called "Search Algorithms for Architectural design" and it presents the mathematical and computational theory involved in the research. The third part is called "Applications of computational Search" and as the name suggests, it presents all of the applications of the concepts explained before.

The **first** part of the thesis takes a look at computational search methods and tools, search algorithms and physical simulation models, all from the point of view of architectural design. There is particular attention centered in looking at historical moments in architecture in which computational or mathematical methods have come into contact with the discipline of architecture. Much has been said about the novelty of digital tools in our field, but there needs to be more information about what is not novel in these methods. This is important if one is to study these methods and try to establish their true impact in architecture, and to try and understand how best to implement them in practice. Many characteristics of these tools have been present in architecture for a long time, from the way we represent geometry, to the very way we explore design possibilities.

The **second** Part looks into the mathematical and computational aspects of this research. The search methods and algorithms used in this PhD thesis are described in detail, discussions on their advantages and known issues are presented as well. The first chapters of this part will look into search algorithms, a general classification, followed by the study of genetic algorithms in more detail.

Multi-objective search will first be discussed in a theoretical way in chapter 7. The concept of contrast is presented together with one of its mathematical representations. We will then look into the algorithms that address multi-objective search, in particular we will study in detail NSGA-II which is the genetic algorithm employed in this research.

The final chapter of this part will look into Parametric Modeling, an example of the parametrization of complex surface geometry is used to explain in detail how to create the model and how to define the possibilities contained in it.

The **third** part of this thesis is devoted to the application of search algorithms in three different disciplines, as well as presenting multi-disciplinary applications. There are four groups of chapters in this part.

The first group presents search algorithms for structural design. This group is made up of chapters 10 and 11. Chapter 10 presents case studies in shell structures, concrete and masonry shells shapes are explored in order to obtain high performance solutions. Chapter 10 is dedicated to load bearing masonry wall structures. This study is mostly concerned with wall openings and thicknesses.

The second group of chapters is reserved for acoustical design of concert spaces. This group is made up of chapters 12, 13, 14 and 15. Chapter 12 is a general introduction to the problem designing spaces intended for the enjoyment of music. A discussion on concert hall design in combination with an introduction to the traditional methods of measurement of acoustical quality. Chapter 13 explains the use of established acoustical parameters for the study of concert auditoria for the search case studies involving them and traditional concert hall types. Chapter 14 presents a study on the use of NURBS geometry in the acoustic simulation of complex shapes. This study was made in preparation to the application presented in chapter 15, this last chapter presents a more innovative acoustical simulation method that is concerned with the early sound in these rooms. In this last chapter, complex free-form shapes are the subject of interest.

The third group of chapters are assigned to the study of energy efficiency in masonry buildings. This group is made up of chapters 16, 17 and 18. Chapter 16 is an introduction into energy efficiency in buildings, the envelope and overall shape of the building, as well as the climates studied in this PhD thesis. Chapter 17 presents a case study on the proportions and orientation of the building, and chapter 18 presents studies on the building envelope.

The fourth group presents the multi-disciplinary search processes developed for this thesis. It is made up of only one chapter describing all of the case studies in multiple disciplines. Two combinations of disciplines are studied, acoustics plus structures and energy plus structures.

Part I

Introduction to Search in Architectural Design

1

Computational search in the early design phase

1.1 The Early Design Phase

It is an intrinsic characteristic of any design process to have phases that have increasing levels of detail. With the evolution of the process, the project needs to be described in more detail and with the appropriate medium. In a traditional architectural design process this is typically represented by a higher scale in drawings, from 1/100, to 1/50, to 1/20 and so on. A higher level of detail means that more and more decisions about the end product need to have been taken. Decisions are usually taken after considering the information available about the consequences, advantages and disadvantages of a particular set of solutions. This information can either come from past experience, studies into the solutions (simulations or expert consultations) or many other sources, but information gathering can be a time consuming and expensive process.

Design problems are almost always approached from a very wide angle in the beginning, the first glance at the universe of solutions has to be a panoramic view in order to consider, explore the highest number of solutions as possible. Designs in this initial phases are usually represented with quick sketches and very little detail is present in these drawings. They go from an *exploratory* phase, to a *development* phase and then to a *definition* phase.

Most design processes are iterative in nature. Steps taken in the process seldom follow a strict sequence, from 1 to 2 to 3. Jumps forwards and backwards on the process are very common. Most importantly for this

research, are the jumps that are taken back to early design. Going back to the exploratory phase happens when designers decide that they need to look further into the universe of all possible alternatives and consider others to the one that they were developing before. These jumps back however, do not mean a complete starting over of the process, because the designer comes back to this point with new information, either information regarding failed attempts and discarded solutions, and/or further information on the definition of the problem itself.

While we can find diverse studies on the design process that seldom agree with each other

Bryan Lawson describes two different "maps"to the design process, the RIBA Architectural Practice and Management Handbook (1965) and the map by Tom Markus and Tom Marver. Lawson points out that from the point of view of the information produced or the output of these maps, they all show a pattern of increasing levels of detail. For example, Briefing, Sketch plans, Working drawings and Site operations in the RIBA map, and outline proposals, scheme design and detail design in the Markus/Marver map. Designers may chose to start with a general view or start by details (selecting materials for example) but the general distinction between the phases remains (Lawson 2006).

In his 1976 article Boyd Paulson was perhaps the first to describe the relationship between design stages, the level of influence in the design and the cumulative cost of the project. He published a diagram (see figure 1.1) of this relationship that depicts a couple of curves describing the inverse relationship, in which, the further along down the design process the lower the level of influence on the project and the higher the expenditures of the project are (Paulson 1976).

Thereafter came many versions of this graph, perhaps the most known version is the MacLeamy curve, but the main concept remains the same. This relationship is a fundamental issue on the importance of the early design stages, and the need for good information and good decisions in this phase.

The early design phase is the moment where the big decisions on the building shape, and many of its defining components, such as structure and distribution are decided.

Building design involves many different technical disciplines apart from architecture. From structural design, mechanical, electrical and hydraulic systems, acoustics, lighting to energy efficiency. Every discipline studies very specific aspects of the building, all looking a different physical phenomena, and evaluating performance values related to user comfort, efficiency and

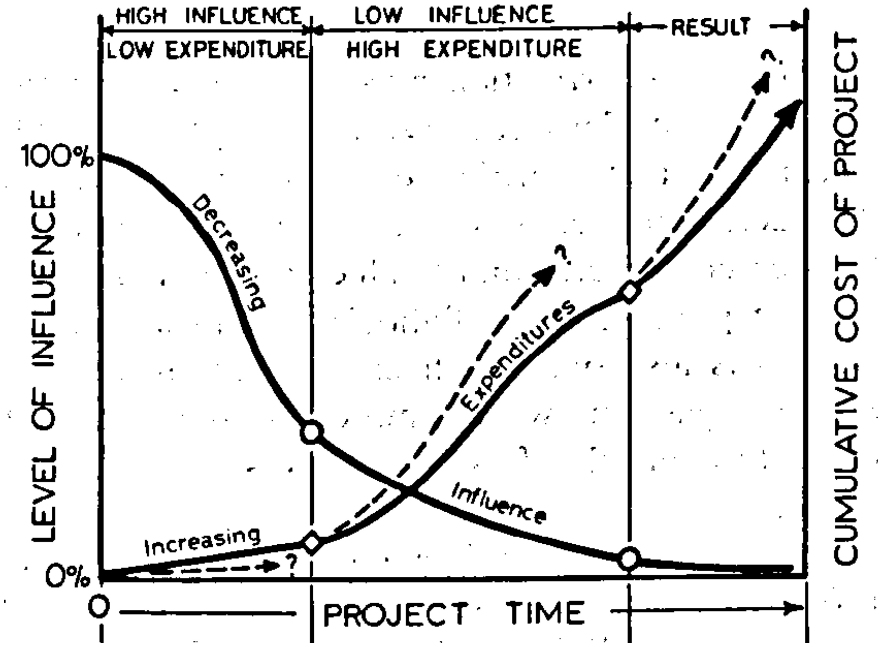


Figure 1.1: Boyd Paulson's curve (Paulson 1976)

quality of the building. All of these disciplines look at specific parts of the building, but most of these parts have functions that involve many disciplines. In fact, "Very rarely does any part of a designed thing serve only one purpose"(Lawson 2006). The façade is studied by lighting, thermal, ventilation and acoustic experts, all looking to improve the design from their point of view. Every component of the building is involved in a multi-disciplinary design process. This is even more evident if we talk about the overall form of the building, its orientation, size and shape.

The early phase of design is the moment when the most disciplines *ought* to be involved, when multi-disciplinary information is most needed.

In such a multi-disciplinary framework it is common to have situations in which the best results in one discipline are achieved by designs that do not correspond to the best result in another one. This issue is clearly described by Christopher Alexander in the introduction to his 1966 book. He describes

an example of a design problem showing contrasting objectives*. He talks about the choice of materials for a household appliance:

> "Time and motion studies show that the fewer kinds of materials there are, the more efficient factory assembly is - and therefore demand a certain simplicity in the variety of materials used. This need for simplicity conflicts with the fact that the form will function better if we choose the best material for each separate purpose separately. But then, on the other hand, functional diversity of materials makes for expensive and complicated joints between components, which is liable to make maintenance less easy. Further still, all three issues, simplicity, performance and jointing, are at odds with our to minimize the cost of materials. For if we choose the cheapest material for each separate task, we shall not necessarily have simplicity, nor optimum performance, nor materials which can be cleanly jointed."

(Alexander 1966)

He accompanies that statement with the diagram shown in figure 1.2. The diagram denotes the four objectives described in the text: performance, simplicity, jointing and economy. The plus or minis signs on the lines connecting the nodes or objectives, are there to signal if the relationship between this objectives is a positive or a negative one. A negative relationship signifies a conflict or contrast of interest between these objectives. In this simple example, alexander only depicts the relationship between simplicity and joining as a non contrasting objective.

These conflicts are very common and problematic in building design. This is especially true in the early design stages when so many different variables and disciplines are involved. When such a large part of the design is still to be decided, it is clear that conflicts have not yet been confronted and solved. This is an intrinsic aspect of this stage of design, and it must be given proper attention if we are to address design tools for the early phases of architectural design.

The resolution of these conflicts can be harder or easier depending on the case. As we will see later in the experimental part of this research, some contrasts are still manageable in the sense that a good trade-off between

*A mathematical description of contrasting objectives by means of Pareto Fronts is explained in section 7.6

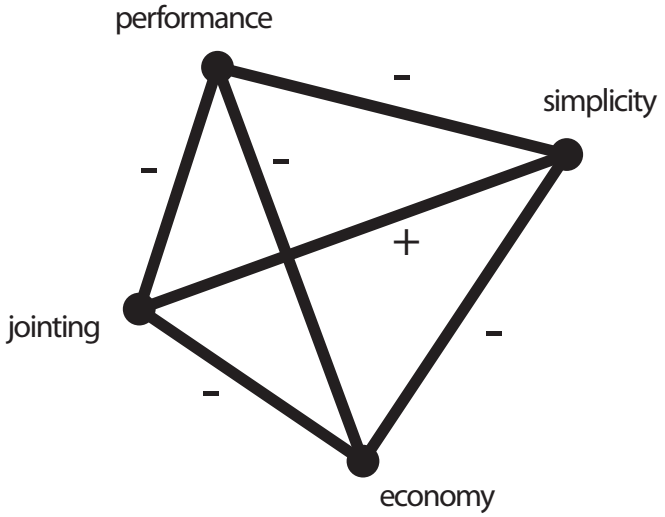


Figure 1.2: Christopher Alexander - Contrasting Objectives Diagram (Alexander 1966)

goals can be found. But other conflicts are harder to negotiate, and in these cases a final decision can only be taken considering more information.

We will refer to these conflicts in design goals as *contrasting objectives*.

If we take another look at Boyd Paulson's curve (figure 1.1) we can see that the potential for design improvement early on is huge, and that in the later stages of design changes are small and limited in scope.

Traditionally, bigger efforts are left for the final stages of the design process, when the technical issues of the project are usually taken into account. Optimization procedures are usually used by a specialist in the field of the particular issue (e.g. a structural efficiency, cost, construction time). However, in the final stages of design the main decisions in most different fields have already been made. When this is the case, it is often too late to make any significant changes in the design and the optimization process serves only as a final definition aid. In the final stages of design the field of possible solutions is very small, when compared to the possibilities in the early design stage. In the early stages of design, the design space is much larger

and so is the amount of information considered in this stage. When we refer to such a final definition process, the correct term is optimization.

The rational and systematic exploration of the space of feasible solutions during the early design stage considering multiple disciplines and contrasting objectives is what we define as a *search* process.

1.2 An Introduction to Search

Hutchison et al. give us the key points to define Search:

- A goal, an objective for the search.
- An Uncertainty about goal location. There can be no search if the goal is directly or easily attainable. Normally, as the search process evolves the this uncertainty tends to diminish.
- The adaptive varying of one's position.
- A stoping rule.

(Hutchinson et al. 2012)

Taking this four points into account we can sketch a very brief definition of Search[†]. Search is the operation necessary to achieve a certain goal when we do not know how to achieve it, by adapting and considering several positions or views about the goal until we either achieve the goal or we stop for another reason.

> "Search - the behavior of seeking resources or goals under conditions of uncertainty- is a common and crucial behavior for most organisms. It requires individuals to achieve and adaptive trade-off between exploration for new resources distributed in space or time and exploitation of those resources once they are found"

(Todd et al. 2012*a*)

The resources we are talking about in search can be varied, but in the present thesis we will be referring to information, either external or internal

[†]A computational and mathematical approach to search is discussed in chapter 5.

information. Most cooomonly this is building performance information (e.g. acoustical quality of a room, energy efficiency of a building envelope). As such the search space is the universe of possible solutions to a given problem, we search for these solutions either from external sources or from our memory of past problems and solutions (internal sources)[‡].

Exploration and Exploitation . An important component in the above definition is the trade-off between exploration and exploitation:

> "Finding a resource typically involves at least two components: an *exploration phase* that investigates possible locations as to where the resource might be located and an *exploitation phase* that involves resource acquisition. Often, the exploration and exploitation phases are not mutually exclusive, as animals often sample and exploit during exploration and continue exploring while exploiting.
>
> Because exploration typically takes away time from exploitation, modulation between the two can be represented as an optimal control problem in which organisms attempt to minimize the time spent exploring for resources but still acquire sufficient information to maximize the resource exploitation. . . More exploration can lead to finding better resources but less time available for exploiting those resources. This trade-off between exploration and exploitation is common to both eternal and internal search problems. "

(Hills & Dukas 2012)

Lawson perhaps characterizes the same exploration vs. exploitation relationship while talking about analysis and synthesis:

> "Analysis as the exploration of relationships, looking in the information available and the classification of objectives. Analysis is the ordering and structuring of the problem. Synthesis on the other hand is characterized by and attempt to move forward and create a response to the problem - the generation of solutions."

[‡]see chapter 3 for a discussion on internal and external search.

(Lawson 2006)

Computation became a very common search method for a large variety of goals, search spaces and disciplines. The power and speed of computation makes for a great tool in any search process, but especially those involving information search, as those that concern the present thesis. The architectural design process is a much more complicated process, one that involves many other tasks apart from information gathering, never the less, a very important aspect of designing is gathering the right information at the right time. This can significantly increase the quality of the end product of the design process. Computational search for the architectural design process is the subject of this thesis, we will look at the specificity of architectural design regarding search, information and the usefulness of this information.

1.3 The "Wicked" problem

In the 1950's, and particularly with a series of conferences in the 60s the field of Design research, or Design methodology was born. A series of books were published by the fields "founding fathers", such as Asimow's "Introduction to Design" in 1962, Alexander's Notes on the synthesis of Form of 1964 and Jone's "Design Methods" of 1970. These seminal works would later be called by Horst Rittel a "first generation" of design methods that had been a necessary but simplistic start to the field. The second generation started with his work on defining what a design or planning problem is, and how it differs from other more scientific problems. In their 1973 paper "Dilemmas in General Theory of Planning", Rittel and Weber in 1973 give us a clear idea of the nature of the design and planning problem:

> " The kinds of problems that planners deal with–societal problems–are inherently different from the problems that scientists and perhaps some classes of engineers deal with. Planning problems are inherently wicked...
>
> ...The problems that scientists and engineers have usually focused upon are mostly "tame" or "benign" ones. As an example, consider a problem of mathematics, such as solving an equation; or the task of an organic chemist in analyzing the structure of some unknown compound; or that of the chess player attempting to accomplish checkmate in five moves. For each the mission is clear. It is clear, in turn, whether or not the problems have

been solved. Wicked problems, in contrast, have neither of these clarifying traits".

(Rittel & Webber 1973)

Rittel and Weber go on to characterize the "wicked" problem:

There is no definitive formulation of a wicked problem. While a "tame" problem can have an unequivocal formulation describing it fully, "wicked" problems are formulated differently by different people, depending upon their understanding of the problem.

Wicked problems have no stopping rule. A mathematical problem can be easily said to be solved, but "wicked" can be worked on or improved almost indefinitely. If we think back to the previous proposition (that "wicked" problems have no definitive formulation) how can we then say when the problem has been solved. Formulation and solution of a problem go hand in hand. Rittel explains that in most cases designers and planners stop working on "wicked" problems not because the problem was considered to be solved in a satisfactory way, but because of other reasons external to the problem. Most commonly designers stop working when they run out of time or money.

Solutions to wicked problems are not true-or-false, but good-or-bad. Scientific or "tame" problems are either solved correctly or incorrectly, theorems are either proven or disproven unequivocally. Solutions to Design or Planning problems can be said to be better or worse from one another, they can be studied and rated, but they are not true or false.

Every solution to a wicked problem is a one-shot operation because there is no opportunity to learn by trial-and-error, every attempt counts significantly. Solutions generally cannot be undone without major investments of time and resources.

Every wicked problem is essentially unique. While "wicked" problems can have similarities and share solution approaches, they are always specific characteristics (for example social, cultural or environmental context, budgets, etc.) that make them all different from one another.

Architectural Design however cannot be reduced to a series of problem solving tasks. As we have seen by the work of Rittel, design problems have intrinsic characteristics that separate them from tame problems. Omer Akin the goes to describe how these problem solving states apply to the ill-defined or wicked problem:

"Types of representations and transformations possible in

well-defined problems are known a priori. Redefinition of these ground rules is not necessary and not allowed. In contrast to this, in design, discovery of new rules is desirable, even though a large set of conventions is available as part of the culture of design. Creative design solutions are often linked to the redefinition of conventional interpretations of design, and creativity is a ubiquitous goal for the designer.

Goal states of design problems are usually inadequately specified at the onset. There are no explicit evaluation functions that can be applied to a state that will result in the unequivocal identification of it as a solution state. Each designer applies his or her own specialized tests to determine wether or not a design is acceptable."

(Akin 1986)

When goal states are not known a priori, fully automated processes are not possible. Automated processes require knowledge on all possible outcomes and directions the automation could or should take. Pre-defined procedures are outlined for all possible scenarios in an automated process. So when we say that in a design process we have not only unexpected results but unexpected goal states, it is clear that design processes cannot be automated. Search processes and other design related tasks can be well formulated and defined for automation. Interaction before, during and after search processes is also an important part of a well defines search process.

1.4 Automation and Design Control

So far we have outlined the concept of an automated computational search process for the early design phase of architectural design. Like with any other process involving automation, the issue of control needs to be considered. Who is in control of the process before, during and after automated procedures is a question to be taken into consideration.

Automation in our daily lives is seen as any process that is done without direct human involvement. This is especially true now that computers are a part of almost any everyday object like a telephone or a car. An automatic car shifts between gears without the driver shifting the gear box with his hands, an automatic cat feeder serves food to a cat without its owner doing anything. But in reality all of these automated procedures were determined

by humans, the car builders established a rule that decides when the car shifts its gears, and the feeding machine is programed by the cat owner to feed the animal three times a day or so. Humans created these processes to do things for them, they defined what the automation does, when it does it and when it does not.

Some automated procedures even leave some decisions up to the system itself. Domotic systems for example decide weather or not to turn light on or off in a space, depending on the presence of users and availability of sunlight. They also heat or cool a space depending on its temperature, as well as various energy saving conditions that the end-user is normally not even aware of. But even in these cases the automated process is carried out only within the confines of what the systems designer defined for all situations.

Automated processes do not presume the absence of rules or definition, quite the opposite. Automated processes follow strict order and procedures defined by the processes author. Computer programs do not define rules or procedures in any scenario, programers do. We can therefore safely say, that any sort of automated process takes place during design, it takes place within the confines of what the designer deemed necessary or desirable.

Search for the early phase of design, as intended in this research, can be characterized as automated processes where the designer is completely involved in the definition of the rules and procedures involved, and where authoriality of the design object is not in question. Designers are also involved in the definition of the representation of the design object during the automated procedure.

1.5 Automation and Representation

Architects do not design by working on the building itself: they create and use different methods of representation and notation. Architectural design processes require representation methods, most commonly based on geometry. Geometry is used to represent spaces and building components. Geometry is expressed through different techniques (e.g. plans, sections, orthogonal projections or 3D models) that allow designers to visualize, edit and communicate their work. Geometrical representation is enriched by shared notations and graphical conventions, perhaps most notably for the purposes of communicating to builders all of the necessary details of the work the must create, the construction documents.

The geometrical representation of space, suitable to being directly trans-

lated into a numerical and algebraic description, is then the starting point for the automated search of design solutions by means of computation for architectural design. A complete description of the design object must be provided for its study. More specifically to search processes, a representation method capable of describing not just one design object but many is required. In fact, an implicit representation must be provided for *all* of the design possibilities that are intended for study.

The representation an entire set of design solutions implies the recognition of common characteristics to all of instances, the features that make each solution part of the desired set related to all other solutions. These common features can be called the *invariants* of the solution set. All solutions in a set are to be similar but not identical and they must share some features and they must be different in some others. Solutions will therefore be described by their common or invariant features and their variable features. We will refer to the numerical quantification of these variable features as design*parameters*.

A parallelepiped for example, is defined by having 6 faces, 12 edges and 8 vertices, and by having straight angles between all adjacent edges and faces. This topology is what makes a parallelepiped, no other characteristics are required for it to be a parallelepiped, it will remain so no matter what is its volume, length or height. The topology of the parallelepiped is its invariant, it defines it as such. The definition of its exact shape is finalized by the use of a combination of dimensions such as length and height or volume and either length or height. These dimensions are the parallelepiped's parameters.

A sphere is defined as a surface that is at any point equidistant to its center, it is not defined by any specific radius or volume, in the same way that all tetrahedron are conformed by 4 triangular faces. These definitions of platonic volumes are well known and serve as clear examples of invariants and parameters, but any topological relationship for any geometrical shape can be represented in terms of invariants and parameters. Dimensions of elements of any topology are common examples, but even topology can be parametrized.

We will refer to this kind of representation defined by invariants and parameters as *parametric modeling*.

Parameters are typically confined to domains of variation, they are only allowed to take values inside a used defined domain. When we use parametric models for search processes, parameters and domains determine the extents of the search space. The number of parameters will determine the dimensionality of the search space, while the domain's shape and range will determine its extension. This is especially important in establishing control

in automated design processes. The procedure mainly controlled trough the definition of invariants and parameters, as well as assessing the parameter domains.

Designers use the concept of *scale* in their design process: larger scales are used to determine overall shapes during the early design phase, while smaller scales focus on details on more advanced design stages. The selection of parameter domains is also subject to the same criteria, domains can be large when general building shapes are studied during a search process, and they can also be small when a more detailed study is necessary during an optimization process.

In automated search, the use of scale implies the concept of *resolution*. Parameter values are chosen inside a user defined domain, but this domain is not necessarily continuous. Domains are usually subdivided into discrete intervals, thus reducing the number of possible values and solutions belonging to the search space. A higher resolution implies smaller intervals in each domain, and thus a higher number of possible solutions, while a lower resolution implies larger intervals and lower number of possibilities. The concept of resolution can be used in a similar way as the concept of scale with the purpose of setting the level of detail in a search process.

1.6 Limitations and opportunities of a computational approach

1.6.1 Computational Performance Simulation

The mathematical description of physical phenomena was developed throughout the centuries, but its large scale application and diffusion was only made possible with the invention of the computer. In the field of mechanics, the well known Finite Element Method (FEM) dates back to the work of Ritz in 1902, but we had to wait until the 1960's to see its dramatic increase in application. Models have been developed to determine the thermal exchanges in the building envelope, the wind resistance of a skyscraper, sun radiation in building facades and roofs or entire city sections, and even the air flow patterns of a sterile operation room. All of these models give designers various performance measures that describe the physical environment they are designing. They all vary in complexity, accuracy, model uncertainty and time consumption.

Performance simulation is a very powerful tool, but *on their own* they are not of too much use to designers, especially in the early design phase.

In most cases, the definition of the building needed to run a simulation is too elaborate for the concept stages. Many decisions have to already have been taken in order to provide the software with sufficient information to operate. This means that the information required to simulate building performance is perceived as too labor intensive for designers to produce in this stage. Commonly they move on from early stages without such information, and only use simulation later on. The output that simulation software provides is not necessarily of any help, especially when the designer cannot afford to consult a specialist for the multiple design solutions that are usually considered in early design. Simulations results are often incapable of helping the designer choose the right path, or make more informed decisions. This scenario often means that in projects with a high technical requirements, for economic and time constraints, the designer is forced to consider very few alternatives, to shorten the early design phase, and to try and make the best of it in later stages of design. In order to introduce computational search in the early design stages, a different approach to performance simulation and data handling is required.

In this PhD research the use of structural, acoustic and energy simulation is discussed and implemented in several case studied. The Finite Element Method (FEM) is used to accurately predict the strains, forces, moments and displacements in building structures of various shapes, materials and loading cases[§]. The acoustic field inside a room is modeled[¶] in an approximate way trough one of the most diffused methods, the so called the raytracing method. Given a sound source, the method can predict the acoustic response in any given point of the room, considering sound reflection, absorption and scattering. Figure 1.3 shows an example of an acoustic study employing raytracing simulation of four different rooms. Energy requirements for heating, cooling and lighting for entire buildings cal also be estimated by the use of dynamic thermal simulation and lighting raytracing software packages[‖].

Simulation accuracy is very much dependent on the accuracy of the data given to the model. In physical problems described by relatively simple models the effect of input data errors on the output can be controlled. However, when the problem involves different physical models, interacting with

[§]A more detailed description of the use FEM software in this research is given in Chapters 10 and 11.

[¶]A more detailed description of the use Acoustic Simulation software in this research is given in Chapters **??** and 15.

[‖]A more detailed description of the use Energy Requirements software in this research is given in Chapter 16.

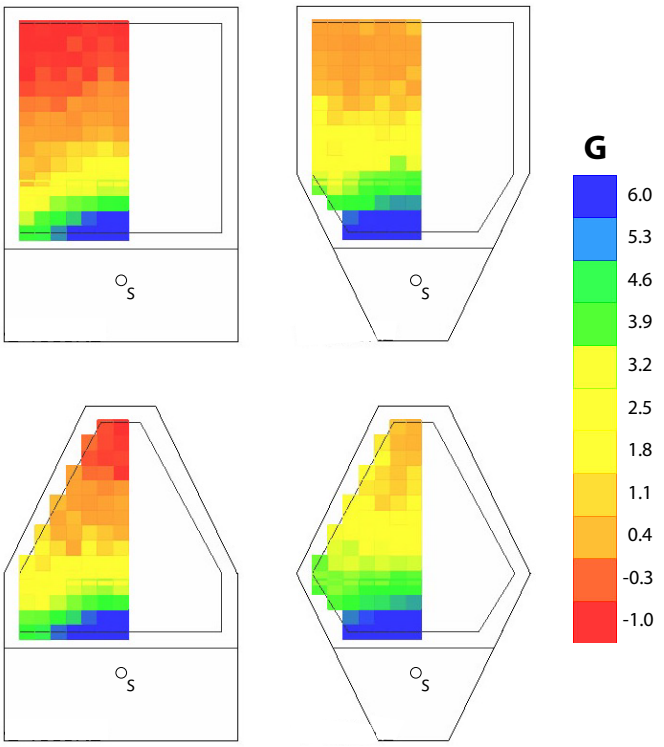


Figure 1.3: Acoustic Simulation Example. Distribution of the Sound Strength Parameter G in four different rooms.

one another in a multi-physics approach, or when non-physical aspects has to be considered (e.g. economic, social, subjective preference), the problem becomes very complex, and the influence of input error on output can increase dramatically. If the data used in the simulation is not correct the results will be completely random or misleading at best. If the phenomena that the designer is trying to investigate are not properly modeled by the simulator employed, the results are equally useless. Designers really need to understand the physical phenomena that are being simulated, the variables involved and how to interpret the results.

1.6.2 Software Customization

Commercial software houses provide an increasing amount of functionality specifically developed for Architectural design. Computer Aided Design (CAD) software has been present even in small architectural practices since the mid 90's. Perhaps the most talked about software tool in the recent years in the architecture and construction community is Building Information Modeling (BIM). BIM can be defined as the use of 3D CAD models in combination with data bases containing drawings, costs, and various characteristics of the building components described in the 3D model. BIM is however a tool that is more used in, and perhaps its better suited for, the design development phase. All of these tools are very successful and are widely used in design studios of all sizes.

Software houses are big companies that are in competition for the architectural software market. They make choices for the development and commercialization of software that are in their best interest, and their interests do not necessarily align with the architect's interests. Software houses are an external entity to the design studio, and their products cannot follow the requirements of each architect or each project.

Large Architectural design firms employ in house software development teams, in charge of creating or customizing software for the firm's projects. Example of these teams include the Specialist Modeling Group at Foster and Partners, the R&D team at Aedas or the Digital Technology Group at Herzog & de Meuron. Some of these teams have pushed the standard for design software all across the industry by collaborating with commercial software houses. However, this has mostly been a reality in very large firms working in large and complex projects with big budgets, but as the commercial software houses are beginning to see the need for software customization and user development, they are starting to create programing environments that are intended to help the architect do just that.

A significant role in this movement has been played by single users-customizers and online communities of that create and exchange customized applications, plugins and scripts that are in turn edited and used by other members of the community. Some of these applications are developed following an Open Source approach. These individuals creatively employed existing computational tools and geometric functions to create customized applications that went far beyond what the software houses were offering. These companies in turn realized the potential of customization and started to include more and more customization capabilities in their products. A significant example of this can be seen in the McNeel's Rhinoceros, the incorporation of the "rhino scripting" environment, and eventually the grasshopper graphical programing environment.

There is an important relationship between the work done by software houses and functionality. The more work software developers do, the less specific or custom functionality is left for the architect. The more work is done by the user-customizer, the more custom functionality he will have. Figure 1.4 shows a diagram of the developer user relationship discussed.

This principle applies not only to architectural software. Considering an example form a different industry, a smartphone application that tells you what the weather is going to be tomorrow just by talking to the phone and asking it verbally, in this case the user needs to make very little effort to get the application to function properly and achieve the desired result. The developer on the other hand needs to make a very big effort. He needs to develop voice recognition software, access weather services and display the output in a meaningful way. We can see this type of user-developer relationship in the center triangle in figure 1.4, we can see that the amount of functionality that the user gets access to is very limited, he can only get to know the weather forecast.

The relationship outlined by the third triangle in figure 1.4 can be seen in the case of the customization of the Catia software by a group of architects in Frank Gehry's office for the Guggenheim Bilbao museum. They took the NURBS functions present in the commercial software and developed tools on top of them to adequately represent all of the complex components of the buildings titanium cladding, among other elements.

When the user becomes a software developer himself he can determine the exact functionality he requires, and expand it as needed. Architects in this category usually use a great deal of available applications and customize them for their own use, often on a project by project basis. This requires not only some programming skills on the part of the architect, but also a good amount of time and effort. When combined with the efforts made by

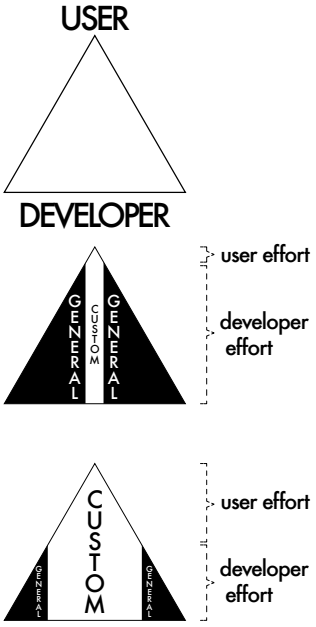


Figure 1.4: User - Developer and Custom use - General use software relationship

software developers, the architects efforts are rewarded by the availability of larger and project specific functionality thats was otherwise unavailable.

1.6.3 Software encapsulating knowledge

Design instruments such as drawing aids and geometrical calculations have always been a part of architectural design practices. They were traditionally close to the architectural profession, being created and employed by architects. The relatively recent arrival of software instruments such as CAD represented a separation between architects and their design instruments. Instruments started to come from other industries, and software tools were made in such a way as to not allow architects to make them their own, they became external to the discipline.

Andrew Witt talks about the inherent (and sometimes unintended) con-

sequences of software development for architectural design from an epistemological point of view. He starts from the distinction between design knowledge and instrumental knowledge in a form that could be applied to the traditional work of the architect.

> "*Design knowledge* is an intrinsic understanding by the architect of formal organization principles such as the relationship of parts to whole, and interrelationships of program constraints, spatial organization, ranges of material effects, and use of geometric methods. It may include disparate and heterogeneous organizational schemes and diagrams. These general principles may be redeployed in various contexts, and need not be tied to particular working methods or automatic tools. In this sense, geometric knowledge is a particular kind of design knowledge: although it may be deductive and procedural, it is not automatic and its application requires a synthetic understanding of design constraints. *Design knowledge* is the most enduring epistemic content of architecture as a discipline, sometimes even hastily equated with architectural knowledge itself.
>
> *Instrumental knowledge* is a more narrow understanding of the procedures to successfully operate a certain type of technology, which would include ability to operate a software, program, script, process, tool, instrument, or machine to intended effect. This is in contrast to the way the term "instrumental knowledge" is used in the epistemology of science, for example: as a description of theories of predictive reliability (and thus instrumentality). Instead, in our sense instrumental knowledge is in fact an intentional knowledge of instrument operation. Instrumental knowledge also enables the creation of systems of interrelated technologies intended to facilitate the aims of design. More generally, instrumental knowledge can include the ability to abstract the inverse constraints of these machines onto design with the aim of pre-rationalizing the design itself, as in the case of drawing machines, fabrication machines, or construction machines. This instrumental knowledge is powerful because it makes procedures encapsulated by the technology in question easily accessible, communicable, repeatable, hackable, and transformable."

(Witt 2010)

In his article "A Machine Epistemology in Architecture. Encapsulated Knowledge and the Instrumentation of Design" he draws a parallel between the geometrical representation machines of the XIX century, such as ellipsographs, and our current use of computational tools for the design, representation and production of buildings. In so doing, he argues that design knowledge has always been dependent of instrumental knowledge, despite the architect's relative disdain and ignorance in the latter:

> "The pervasive use of digital technology in the conception and execution of buildings dramatically increases our reliance on representational and operational systems of which we have incomplete understanding but that we nevertheless trust implicitly... The machine, particularly the computer, calls into question the self-understanding of architecture and its self-imposed alienation from technical processes. There is a strong tendency, arguably beginning with Alberti, to dichotomize design knowledge and instrumental knowledge, and to relegate technical or mechanical expertise to the domain of specialists or operators. Perhaps this can be explained by a mistrust of the architects need to rely on mechanical, electrical, computational, or conceptual operations of which the architect cannot have complete understanding. This trust in machines, however, far from being an innocent conceit, represents an implicit belief in the possibility that collective memory and design knowledge can be instrumentally encapsulated in machines. It represents not a barrier to advancement of architectural knowledge but a great opportunity."

(Witt 2010)

Witt's enthusiasm for the advancement of design knowledge through encapsulated instrumental knowledge could reach the design community in a stronger way by emphasizing the fact that when the architects are the impulse and creators of instrumental knowledge, the relationship between design and instrument is stronger. Architects should not completely conceit defeat in the generation of instrumental knowledge, but be at its forefront. This remark can be understood in the context of the user development relationship outlined above and in figure 1.4. The more the architect takes part in the creation of the software instruments he uses, the more will he profit from its functionality, and the better are the chances for him to make advancements in design knowledge.

Most contemporary software instruments used by architects come from distant fields. From computer graphics and animation to civil and aeronautical engineering, many distant disciplines have contributed to the architect's computational toolbox. This would explain the separation between design and instrumental knowledge Witt is describing. But the customization of software is an important instrument being embraced by the architecture community, and represents an opportunity to regain instrumental knowledge in the hopes of further advance our discipline.

In this process of instrument creation and encapsulation by architects, the dichotomy described above could tend to disappear, design and instrumental knowledge become almost indistinguishable in architectural practice. We can certainly say that design knowledge can be encapsulated in software or machines, facilitating their use. But we can also say that the use of encapsulated knowledge and design software tools such as those proposed in this PhD can further generate design knowledge. Information obtained during search processes, if carefully studied, represents an advancement on design knowledge.

1.7 Interactivity and Search

> "Take an optimization model. Here the inputs needed include the definition of the solution space, the system of constraints, and the performance measure as a function of the planning and contextual variables. But setting up and constraining the solution space and constructing the measure of performance is the wicked part of the problem. Very likely it is more essential than the remaining steps of searching for a solution which is optimal relative to the measure of performance and the constraint system".

(Rittel & Webber 1973)

In this 1973 quote, Horst Rittel warns us on the risks of using optimization (or search methods) for design problems. The wicked part of design and planning problems is very often not solvable by means of computation. Many aspects of design problems can be addressed by performance measures, but many cannot. Designers are therefore required to to ask the right question in such search processes, to formulate the problem correctly. When formulation of search processes is done successfully, computation can pro-

duce information, performance data and even propose favorable solutions to the formulated problem. The input from the designer throughout the automated process is fundamental.

The most important moment of interaction between the designer and the search process is of course the formulation of the problem: the definition of the parametric model (invariants and parameters), the selection of the performance criteria and the simulation method. It is in this moment when the designer has the most control over the search process and consequently when he can make the biggest mistakes.

There are other moments when the designer can interact with the search process. Christian Derix talks about this issue and the importance of interaction in his 2010 paper "Mediating spatial phenomena through computational heuristics":

> "Wicked problems like layout or urban design require the experience of designers to negotiate the many explicit and implicit aspects that can be represented through computation. Particularly, when design aspects are not discursive and the amount of data is large, the key organizing principle of designers and design teams are their learned heuristics, not performance indicators and data sets. While computation shouldnt imitate analogue heuristics, it can express its own search mechanism via visualization of processing steps. If a designer can interfere with computational heuristics and observe the search struggle, the opportunity identification between designers analogue and computational heuristics are given. This enables the validation for wicked problems when no explicit goals are set."

(Derix 2010)

The information produced by and during the search process can be viewed by designers and involve them directly, by interacting and modifying the process itself. There seems to be a limitation in the participation of the designer with regular search methods, that push their use to later design phases. On the other hand, interaction, the ability to provide input in real time, introduces a real participation that is essential for the early design phase. The ability to influence the result by means of interaction, is evidence of design, not optimization. Search processes are design oriented, not just performance oriented. Not all requirements or design ambitions are introduced as performance criteria, this is left to the discretion of the

designer. This does not mean that those design requirements and intentions are not a part of the search process, by means of interaction the designer is actively introducing them in the process. The search process would then have explicit and implicit design goals. From a performance point of view, the final result may therefore not be an "global optimal" solution, but it will be what the designer wants, a better informed design.

The opportunities of interaction in search algorithms are different depending on the algorithm itself. But the general idea is that the very formulation of the algorithm (or parts of it) are modified during the execution, responding to the designers evolving intentions. As the search process evolves, and the knowledge acquired by the decision maker is increased, implicit or explicit design goals can vary and the search process can be steered on a new direction. The most obvious way of doing this is by changing the parametric model itself, either by adding/subtracting variables, or by changing their domains. Changing parameters or domains, without modifying the design goal of the search process has no negative effect on the comparison of the resulting performance values. While the parametric model that generated the solutions changed, the performance criteria remained untouched, hence we can still compare solutions before and after the interactive modification.

Another way of doing this is by changing the design goal itself. It can be slightly modified to better suit the problem, or to try and manipulate the solutions being generated. This operation is perhaps counter intuitive when compared to the one described above. It has the disadvantage of producing results that are not comparable to each other. Results with different design goals cannot be compared on terms of their performance values.

The data produced during these interactions can be stored in a "search tree". A sort of Search history of the decisions made during the process, the interactions, modifications and performance values. Figure 1.5 shows a diagram of a search tree, in which the designer after 4 search iterations was presented with solutions a, b and c. The designer selected solution a (signaled by the letter A in caps) and after a few more iterations solutions d and e were considered, solution E was selected, and so on. After 13 iterations the designer ended the search process. As the diagram suggests, the data pertaining to all solutions and iterations is saved in the computer, enabling the designer to restart the process at any point, or to consult the data later on.

One of the main concepts that allow architects to engage the whole scope of the project in the early design phases is the concept of scale. Designers often use larger scales in these phases so that they can concentrate on the big picture, and make faster decisions on the vast majority of the building's

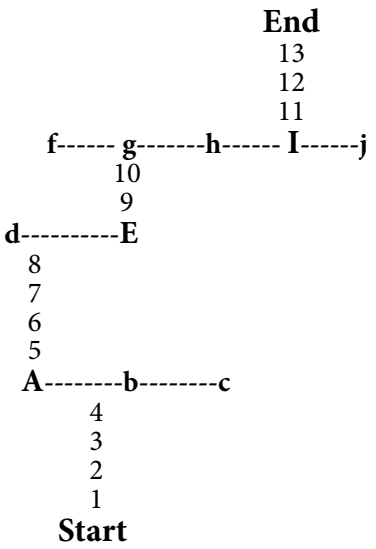


Figure 1.5: Search Tree diagram

form. Slow interactions with simulation models and search algorithms, mean that the designers need to leave these tools for later design. Calculation times are important when it comes to real-time interaction.

As previously introduced, the concepts of scale and resolution can be applied to Performance based Search in order to reduce calculation times by limiting the size of the search space and facilitate interaction. Like architects often do, search methods can first work with low resolutions in their parameter domains. This means discretizing the search space in a coarse way, effectively reducing the number of calculations needed to provide the designer with an broad idea of where the search process is leading him. This means that the designer would be utilizing broad strokes in the beginning and progressively improving the resolution as he focuses on specific areas, performances, or geometries. This can be an effective way to engage the designer and improve interaction.

An alternative way of interacting with the process is one that takes place

after the calculation is done, a sort of data mining of the search output. This mode of interaction is applicable when large search spaces are involved, resulting in large data sets. Interaction in this case cannot modify future results since all calculation is done, but it can be the starting point of a new search process involving new information, a new iteration with more information that can help better formulate a new process.

The Early Design Phase is about immediate restitution, real time interaction, the way a pencil gives an immediate result. Computational search is not immediate, depending on a variety of issues, a search process can take minutes, hours, even days. Interactions during the design search imply that the designer is constantly involved in observing and modifying the process. Thus the process should be producing feedback constantly, in real time if possible. This is not always possible.

2

Algorithms and Parameters in Architectural representation

Architects create and use different representation methods during the design process, they conceive, edit and study their buildings through some form of representation. Representation is also used to transmit design information to builders. Different media have been used by architects throughout history: The spoken word, the written word, the handmade drawing, the printed drawing, the scale model, the digital file. Some media are inherently more inductive towards one representation method over another.

Architectural representation has been addressed throughout history by architects in treatises and manifestos, It has been the subject of study in this discipline for along time. Andrew Witt considers it design knowledge and not instrumental knowledge, which is not surprising, since representation has always been considered a part of the architect's fundamental techniques.

The type of architectural representation described in this PhD thesis is hardly new, parametric models, as well as their iterative nature and relationship with the design process can be traced back to previous centuries. An understanding of the roots of these methods of representation seems to be an important step in relating existing design practices with automated processes such as search.

2.1 The roots of Parametric representation

Architectural scaled Models have been present since the very beginning of western architecture. There is evidence to suggest that wax and wooden

models were used by greek architects. there is some debate as to the exact purpose of these models, if they were for exhibition, survey, building or design purposes. It is also unclear the role they had in roman times, but it seems clear that for Vitruvius they played no part in the ideation of the building.(Scolari 2012, Carpo 2011)

The purpose of making models in architectural design is perhaps clearer in Alberti's treatise *De re aedificatoria*. Massimo Scolari describes Alberti's ideas for the use of Models:

> "His model is an instrument of experimentation and reflection with which to ascertain the buildings's structural stability, its orientation, the layout of the main walls, and the adequacy of its roofing. It is used to try out the most likely solutions to each single problem and to make a precise calculation of the costs of the work... the model should be "nudos et semplices", crafted from simple materials so that it is the architect's true conception that emerges, rather than the skill of the model maker."

(Scolari 2012)

The words "Experimentation and reflection" denote the fact that, for Alberti, models were very much a part of the design process and were not involved in the construction process. The model was not entrusted with a notational aspect as other means of representation. In Alberti's writings, orthogonal Drawings were the preferred method of representation when it came to construction. The iterative nature of design and design representation is also present on renaissance model making, Massimo Scolari quotes Filippo Baldinucci from his "Vocabolario toscano dell'arte del disegno" from 1681:

> "The first and most important task in the making of the work is the model, since it is by means of trying out his ideas and *altering* them that the Artificer arrives the most beautiful, most perfect solution. "

The fact that both buildings and scaled models are tridimensional physical objects has always been a great advantage for the model, in the sense that the information is conveyed more directly, a tridimensional object is represented by a tridimensional model. In contrast with orthogonal drawings, models present themselves as direct scaled copies of the building. The

only abstraction left to the viewer is the scale and size of the object in relation to human scale.

It was not until very recently that scaled models were given notational importance, and this only happened because of fairly recent technology. A process called 3D capture was used in the design of the Guggenheim Museum in Bilbao, in which a 3D model was "digitized" by means of computer sensors originally developed for medical purposes. The exact dimensions of the model are taken into the computer where further design and detailing is done, before using the computer to make a new physical scaled model for Gehry's inspection (Marshall 2001).

From scaled models, parametric models inherit their iterative design nature, their use as experimentation and analysis tools, and their tridimensionality.

Modern descriptive geometry defines orthogonal drawings as being projections from a point situated at infinity, meaning that parallel lines do not cross each other as they do in perspectival drawings (or rather that they meet at infinity), and therefore lines retain their dimensions throughout the drawing. It is for this reason that Leon Battista Alberti in his treatise assigns them the role of the notational documents to be used by the builders. He makes a clear distinction between representations suitable for design and reserves for construction only that can be measures with precession (plans and elevations).

In Alberti's conception of the architectural design process, the architect is not to be involved in the construction process. Therefore the information necessary to build his design must be very well described and detailed enough for the builders to proceed in his absence. As the distance between designer and builder grows, the need for accurate notational tools also grows. Architecture becomes more and more allographic. This is why notational orthogonal drawings become highly important in Alberti's idea of Architecture.

> "Alberti's design process relies on a system of notation whereby all aspects of a building must be scripted by one author and unambiguously understood by all builders. Its principal notational means reside in the scaled a measured drawings of plans, elevations, and side views defined in the second book of *De re aedificatoria*."

(Carpo 2011)

Alberti sets the basis for the modern design and construction process and even with the advent of computational CAD, CAM and BIM technologies, we still require orthogonal plans and sections for most construction processes, especially in the bureaucratic and legislative review stages.

It is widely known and documented that the dimensions of structural and ornamental features of classical architecture are strictly related to each other by means of a system or proportions. Bernard Chache explains the presence of parametric relationships in these proportions in Vitruvius' *De Architectura*:

> "Let us turn again to the oldest treatise on architecture that has come down to us. Its author, who adorns himself with the title of "architect", spent the greater part of his career designing machines of war. The components of these machines were assembled according to parametric relationships. The most important of these relationships one far more complex than any simple fraction served to determine a module that was dependent upon the weight of the stone that was to be catapulted. Invented for the purpose of calculating this proportion was an apparatus (the Greek word for it was armonia), constructed of wooden slats and a cable, a device which, while not a computer in the contemporary sense, nonetheless facilitated the execution of a large number of computing operations. Such contraptions must appear familiar to contemporary architects who design components that can be varied in dependency upon parametric relationships. Is it inconceivable to construct a trajectory of tradition between todays parametric design techniques and the oldest surviving architectural treatise?
>
> To be sure, one should guard against excessively hasty comparisons. The contexts of antiquity cannot be equated without further ado with the circumstances of our own times. Still, it would be an error to consider such historical contexts in strict isolation from one another, since that would eliminate at the outset all questions regarding the survival of related problematics."

(Cache 2009)

Chache illustrates the presence of parametricism using Vitruvius' writings on war machines, not on buildings. The reason for this might be that parametric relationships in these war machines served a design process, the construction of the machine was directly related to the parameter (the weight of the stone to be catapulted). This ties function and design very tightly through a parametric process. But surely parametric relationships in antiquity was not confined to the design of war machines. Mario Carpo describes parametric processes involved in the architectural order:

> "The classical columnar system, first described by Vitruvius and later known as "the rule of the orders", is based on precisely determined norms and standards. Every component has a recognizable form and a name; composition - the assembly of the parts of the system - follows rules similar to those applied by ancient rhetoric by literary discourse. The precise quantification of particular dimensions and distances was an essential part of this process. Most normative measurements were defined as proportions: the traditional units of measurements were derived from parts of the building itself - typically, but not exclusively, from the diameter of a column."

(Carpo 2003)

Although Alberti was a strong believer in orthogonal drawings as the notational tool to instruct builders, he knew that for the diffusion and reproduction of his treatise he needed a different method of representation to describe his versions of the classical orders.

> "Before the invention of print, manual copies of drawings were famously untrustworthy, and as a result, images were seldom used, or altogether avoided, whenever precise copies were required. In such cases, non visual media (alphabetical or alphanumerical) were deemed safer. For centuries in the classical tradition (from antiquity to the middle ages to the early Renaissance), most architectural descriptions were verbal, not visual."

(Carpo 2011)

In book III of *De Architectura* Alberti writes about Vitruvius' description for the Attic Base, while in book VII of *De re aedificatoria* he writes his own instructions for the same base. In his paper "Drawing with Numbers"

Mario Carpo explains some of these descriptions in detail and how they evolved during the course of history. Alberti's instructions for calculating the proportions of the Doric base are translated and explained as follows:

> "First, take the diameter of the column at the base,and divide it into two equal parts. This gives the total height of the base. Then, take this segment and subdivide it into three equal parts; the lower third is the plinth. The take what is left, divide it into four equal parts; the upper quarter is the upper torus. Then take what is left and divide it into two equal parts; the lower half is the lower torus. Then take what is left and divide it into seven parts, and the upper and lower seventh are the two fillets. What is left is the scotia, sandwiched the two fillets and tori. Thus the sequence is completed."

(Carpo 2003)

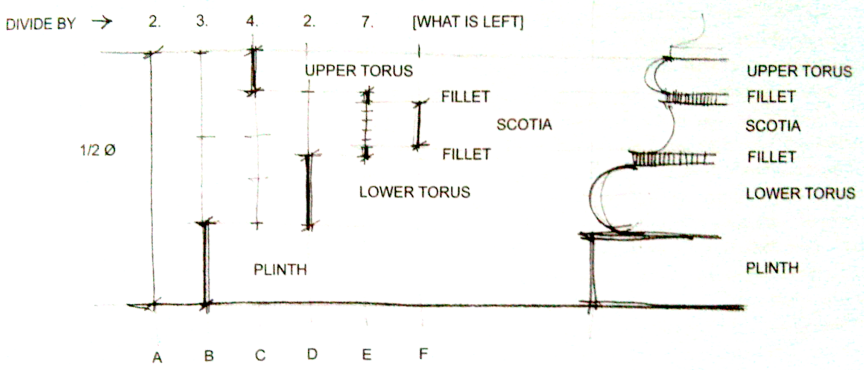


Figure 2.1: Mario Carpo's diagram of Alberti's instruction for determining the proportions of his Doric Base in the seventh book of *De re aedificatoria* (Carpo 2003).

These verbal descriptions or instructions that Carpo is talking about can be best defined by the term *Algorithm*. Figure 2.1 shows Carpo's diagram of the procedure and the resulting proportions. They are a sequence of mathematical operations by which we unequivocally obtain all measurements in the base, starting form one given dimension, in this case, the diameter of

the column at the base. The relationships between parts of the base are invariants, they do not change as the diameter of the column does. The diameter is the only number that is not obtained by the algorithm. If we think of this number as a variable number or designer defined *parameter*, this process could be called parametric modeling. It is the same exact operation used by contemporary designers with different geometries, parameters and design goals in mind.

Before Alberti, and in some cases after, this kind of algorithmic representation was used also as notation in the building site. The operation can be made in a geometric way with the use of a ruler and a compass. The algorithms described by the architect were reproduced live in the building site to obtain the desired geometry.

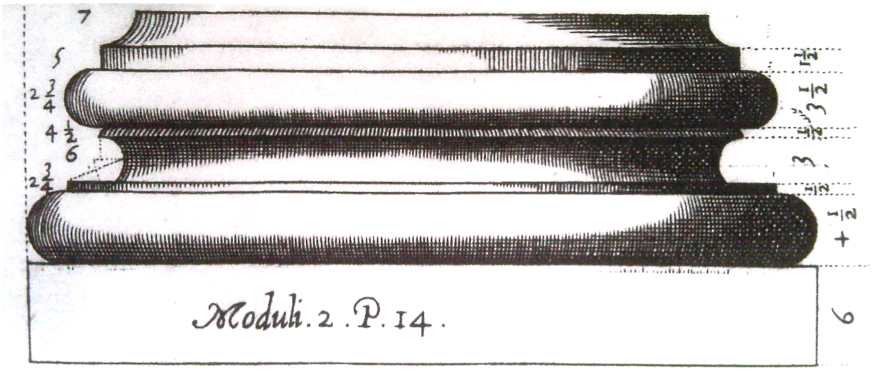


Figure 2.2: Attic base from Giacomo Barozzi da Vignola "Regola delle cinque ordini" (Rome, ca. 1562-63) (Carpo 2003).

Carpo goes on to write how after Alberti, and more importantly, after the invention of the printing press that could reliably reproduce images, verbal or textual descriptions of geometry started to change. Images started to appear, and proportions started to be expressed geometrically, then in fractions, and finally in numbers. Figure 2.2 shows Vignola's Attic Base. For the first time, proportions were pre-calculated (leaving only a few fractions) by the author (Vignola) and the reader simply used the numbers as dimensions, much like we use today in contemporary orthogonal drawings. The numbers in the image represent the number of "moduli" belonging to each measurement.

2.2 Contemporary Parametric representation

Computer Aided Design has its roots in the 1950's, but it only started being a part of the average architecture studio in the mid 90's. And even then, most contemporary architecture studios make use of this technology in a very traditional way. Digital Orthogonal drawings and 3d Models are used in a way that is not very different from what Alberti described, the only significant improvement the new media brought is that it allows the architect to interact with the building and redraw it with higher speed and precision and with lower costs. The arrival of CAD technology represented an improvement in speed ad reliability.

Non Rational B-Splines or NURBS were an important addition to the architect's representational toolbox. Complex Curves and curved surfaces have always been present in architecture, but this innovation on the mathematical description of this geometry brought with it precession, repeatability and ease of use the likes of which had never been seen before. This mathematical application gave architects the notational capability of describing very complex shapes.

While the arrival of these technologies might not have had a strong importance at first, they would pave the way for contemporary parametric modeling. CAD software is fundamental in the revival of algorithms as a method of architectural representation, and most importantly, their use as contemporary design tools. The power of computation is now beginning to be used for design purposes, not just representation.

Algorithmic representations in classical and renaissance treatises are, from a mathematical point of view, very similar to contemporary parametric models. The algorithms used by Vitruvius and Alberti were perhaps mostly used as geometric representation, is it arguably unlikely that they were used as design methods. Changes in the classical orders (hence on the algorithms) were not made by architects from one building to another, much less in a single building. So we can say the algorithms were not used as parametric models with design purposes but only as geometrical representation, their use was strictly notational *.

Perhaps the most significant difference between pre-computation and present use of algorithms is their use of CAD software as an immediate

*There are some geometrical an proportional constructs that were used to make geometry and structural elements parametrically to the room size (see (Tomasoni 2008) pp. 37-40 for examples on vault scaffolding construction in the XIX century), but these examples are hardly comparable to the use that is being seen today. Also Bernard Cache writes about the use of parametric models in Vitruvius' war machines.

graphical restitution so that the designer can quickly interact with the resulting geometry. This interactive aspect is a fundamental part of contemporary parametric modeling, because by interacting and modifying the model in real time, the designer maximizes the number of design iterations and hopefully improves the quality of the building in some way. While not perfect[†], parametric modeling is currently the fastest and most flexible geometrical representation method at hand, especially when compared to traditional drawings and scaled models.

This flexibility is not only exploited from a design efficiency point of view, different instantiations or "versions" of the model are often used in the same building or object:

> "A series is a framework of parameters designed by the architect, within which a variety of design versions may be realized. Each of these design versions is unique and yet also part of the series. The parts assembling each of the series' designs are no longer necessarily mass-produced but could rather be mass-customized... Versioning is at the core of the digital form itself; its signature and its authenticity derive from the parameterized repetition which give computer-generated design its characteristic combination of tightly disciplined structure and formal variability. Its not just the new calculus-powered curvaceousness, which is characteristic of a digitally informed age; it is also a disciplined groundwork of order that underpins the whole operation – the rhythm of a powerful Turing Machine that drives the versioning at the heart of the digital aesthetic."

(Rocker 2008)

Discipline and order are indeed required in the definition of a parametric model. To determine what is invariant, what needs to be constant throughout the entire series, is to determine what is fundamental and important in the design process.

The architectural practice today is characterized by its relentless speed. Design and construction must both be done at an ever growing speed that reduces the designer's ability to carefully consider all of the options available to him, and the consequences of his choices, often at the expense of the

[†]See (Davis 2013), in chapter 2 of his PhD thesis he explains the seldom discussed challenges of creating completely flexible parametric models with the current tools available to architects.

resulting buildings quality. A very important aspect of computational design tools to provide the designer with a geometrical representation method that is fast and flexible enough to accommodate this fast and demanding building environment. Higher flexibility and speed allow designers to consider a higher number of alternatives, edit their designs with much less effort (Davis 2013). Flexibility and speed are thus a very important advancement in representation, not just a slight improvement, it is the response to todays fast design and construction cycles.

Parametric modeling has also represented a bridge in the gap between design and construction. This representation method allows architects to describe a large amount of different objects in a precise and fast way:

> "If all that is built is built from notations, and if the drawings (or models) must contain all of the necessary data for an object to be built identically to its design, it follows that in most cases what can be built is determined by what can be drawn and measured in drawings. And as the notational system that encodes and carries data in architectural design is mainly geometric, it also follows that the potency of some geometrical tools determines the universe of forms that may or may not be built at any given point in time... This notational bottleneck was the inevitable companion of all allographic architecture from its very start... By bridging the gap between design and production, this mode of digital making also reduces the limits that previously applied under the notational regimes of descriptive and pre-descriptive geometries, and this may well mean the end of the notational bottleneck"

(Carpo 2011)

The notational bottleneck Carpo is referring to is at the heart of contemporary computational design and construction tools, and parametric representation of geometry is a key aspect. Moreover, we are increasingly reducing the difference between design representation and building notation, until eventually no bottleneck will exist.

In his book the "Alphabet and the Algorithm" Mario Carpo talks about algorithms that were used by designers to describe building components, and that those same algorithms were repeated by workers on the building site. This statement does not apply in todays machine production. Computer controlled mills or 3D printers do not follow the same codes that designers

use to generate their models. However, parametric models are an ideal environment for the representation of construction components and their assembly, regardless of their means of production, be they mass produced or customized.

The term mass-customization is generally applied to automated construction processes that are capable of producing components all different from each other with no additional cost. Mass-customization is comes as a response to mass-production in which the costs of molds and machinery was amortized by the production of many identical pieces. But mass-customized components only make sense when we have the means to design all of these different pieces in mass as well. The description of components interns of invariants and parameters allows designers to maintain rigorous control over all of the different pieces without manually drawing each one.

The algorithms that define parametric models are not described in verbal or written form as they were in Alberti's time, they are described in computer programming environments. There are multiple programing environments used in the design community today. They vary in their potential functionality, user interface and computation times. Initially parametric models were always done by incorporating customized pieces of software or "scripts" inside CAD environments. These were written by the user himself in various programing languages, sometimes adapted by the CAD programs to simplify their use. Examples of this are "Rhinoscripts" in Rhinoceros© and Maya Embedded Language (MEL) scripts in Maya©. Recently, commercial CAD software have been expanded to include graphical programing environments that enable users with little or no programing skills to create parametric models. Examples of this category in the architectural design community Generative Components© from Bentley Systems©, Grasshopper© for Rhinoceros© and very recently Dynamo© for Autodesk Revit©.

Designers can define a whole family of geometrical objects, to study in a search process, inside a single parametric model. However defining a parametric model is a process that needs to be made carefully from a design point of view[‡]. Designers must be aware of all of the possibilities of that particular parametrization, all of the geometry that is included and all of the geometry that is not included. This careful study of the possible outcomes of the model is fundamental when eliminating or considering possibilities.

[‡]An example of the consequences of different parametrizations of a given geometry, but from a search point of view is given in section 11.2.